Une approche par dissimilarité pour la caractérisation de jeux de données

William Raynaut
IRIT UMR 5505, UT1, UT3
Universite de Toulouse
william.raynaut@irit.fr

Chantal Soule-Dupuy
IRIT UMR 5505, UT1, UT3
Universite de Toulouse
chantal.soule-dupuy@irit.fr

Nathalie Valles-Parlangeau
IRIT UMR 5505, UT1, UT3
Universite de Toulouse
nathalie.valles-parlangeau@irit.fr

Cedric Dray
INSERM, U1048
Universite de Toulouse
cedric.dray@inserm.fr

Philippe Valet
INSERM, U1048
Universite de Toulouse
philippe.valet@inserm.fr

ABSTRACT

La caractérisation de jeu de données reste un verrou majeur de l'analyse de données intelligente. Une majorité d'approches à ce problème agrègent les informations décrivant les attributs individuels des jeux de données, ce qui représente une perte d'information. Nous proposons une approche par dissimilarité permettant d'éviter cette agrégation, et étudions son intérêt dans la caractérisation des performances d'algorithmes de classification et dans la résolution de problèmes de méta-apprentissage.

Keywords

Caractérisation de jeux de données, Dissimilarité, Méta-attributs, Méta-apprentissage, Sélection d'algorithmes

1. INTRODUCTION

L'émergence du phénomène de données massives crée un besoin grandissant en analyse de données, et bien souvent, cette analyse doit être conduite par des experts de différents domaines ayant peu d'expérience en science des données. Afin de leur permettre de tout de même exploiter efficacement leurs données, divers travaux ont proposé des méthodes d'assistance intelligente à l'analyse de données [31, 20]. La caractérisation de jeu de données, problème apparu avec les premières ébauches de méta-apprentissage [5], constitue encore l'un des verrous majeurs de l'assistance intelligente à l'analyse de données.

Dans le cadre général du méta-apprentissage, le problème de caractérisation de jeu de données consiste en la définition d'un ensemble de propriétés de jeu de données (ou méta-attributs) permettant leur caractérisation précise qui doit de plus être utilisable par des algorithmes de méta-apprentissage. Afin de se conformer aux prérequis de la plupart des algorithmes de méta-apprentissage, ces propriétés sont généralement agrégées en vecteurs d'attributs de taille fixe, ce qui peut représenter une importante perte d'information [10]. Nous étudions la possibilité que les limitations des techniques courantes de caractérisation de jeu de données soient l'un des obstacles majeurs à la bonne performance de la sélection d'algorithmes. Nous nous concentrons en particulier sur la définition d'une représentation des jeux de données permettant d'utiliser toute l'information disponible pour leur caractérisation.

Tout d'abord, la section 2 présentera les approches existantes de caractérisation de jeux de données, et illustrera sur un exemple la perte d'information provoquée par l'agrégation d'attributs. La section 3 s'intéressera aux fonctions de dissimilarité, en étudiant les propriétés désirables pour la caractérisation de jeux de données, et présentera une fonction candidate. La section 4 étudiera ensuite la validité de la dissimilarité proposée d'un point de vue théorique puis expérimental. Enfin, la section 5 présentera nos conclusions et discutera de potentiels développements.

2. APPROCHES PRÉCÉDENTES ET MOTIVATION

Le problème de caractérisation de jeux de données a été étudié selon deux axes :

- Le premier consiste en l'emploi de mesures statistiques et information-théorétiques pour décrire le jeu de données. Cette approche, notamment mise en avant par le projet STATLOG [15], et employée dans une majorité d'études postérieures [25, 9, 13, 21, 14], présente nombre de mesures très expressives, mais sa performance repose intégralement sur l'adéquation entre le biais de l'apprentissage effectué au méta-niveau et l'ensemble de mesures choisies. On note parfois l'emploi de techniques de sélection d'attributs à ce méta-niveau [11], mais les résultats expérimentaux ne permettent pas de conclure à la supériorité de quelconque mesure indépendamment du méta-apprentissage employé [22].

BDA 2016, 15 au 18 Novembre, Poitiers, France.

- Le second axe d'approche considère quant à lui non pas des propriétés intrinsèques du jeu de données étudié, mais plutôt la performance d'algorithmes d'apprentissage simples exécutés dessus. Introduit comme "*landmarking*" par [19], cette approche emploie initialement le taux d'erreur d'un ensemble d'algorithmes basiques comme métadonnées. Comme précédemment, les résultats suggèrent une forte dépendance de l'efficacité de cette approche avec le choix des algorithmes de base et du méta-niveau, ne révélant aucune combinaison uniformément supérieure. Des développements postérieurs ont introduit des mesures plus complexes, tel [18] proposant comme méta-attributs des propriétés structurelles d'un arbre de décision construit sur la donnée. Les expériences conduites par [4] sur ces différentes approches tendent à conclure que toutes peuvent réaliser de bonnes performances dans diverses parties de l'ensemble des jeux de données, sans qu'aucune ne domine globalement.

Le problème de caractérisation de jeux de données a donc déjà reçu une certaine attention dans le domaine du méta-apprentissage, mais l'agrégation des méta-attributs en vecteur de taille fixe y reste une constante. Cette agrégation représente cependant une importante perte d'information, que certaines approches ont déjà tenté de limiter, notamment par l'utilisation d'histogrammes [8]. On peut illustrer ce problème sur l'exemple suivant.

Exemple *Considérons deux jeux de données,* $\boldsymbol{A}$ *et* $\boldsymbol{B}$ *illustrés en figure 1.* $\boldsymbol{A}$ *décrit 12 attributs de 100 individus, et* $\boldsymbol{B}$ *10 attributs de 200 individus. On souhaite comparer les résultats de 5 mesures statistiques et informationnelles relevées sur les attributs individuels de ces jeux de données (comme illustré sur le second attribut de* $\boldsymbol{A}$*).*

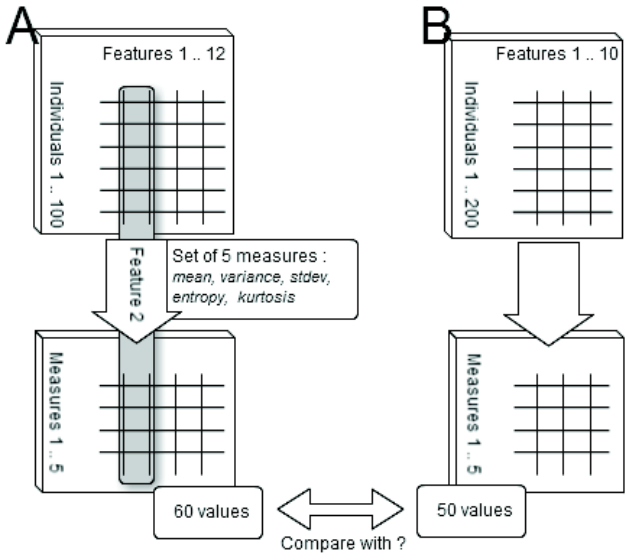

Figure 1: Propriétés d'attributs individuels

L'information complète que l'on souhaite comparer est donc un vecteur de 60 valeurs pour $\boldsymbol{A}$ *et de 50 pour* $\boldsymbol{B}$*. Une approche classique [8, 27] serait de faire une moyenne de chaque méta-attribut selon les différents attributs des jeux de données, perdant ainsi l'information caractérisant individuellement chaque attribut (Figure 2)).*

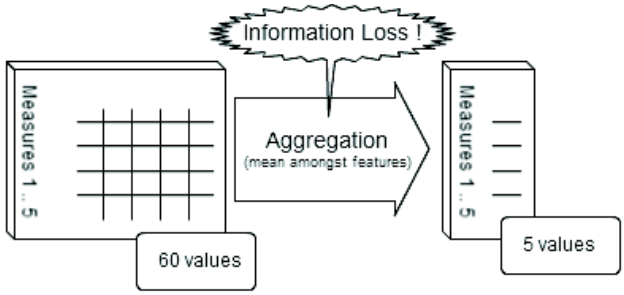

Figure 2: Moyenne sur les attributs

Notre approche est de comparer les attributs de $\boldsymbol{A}$ *et* $\boldsymbol{B}$ *par paires les plus similaires, comparant les attributs en surnombre à d'hypothétiques attributs vides. L'hypothèse émise ici est qu'un attribut absent équivaut à un attribut dont aucune valeur n'est connue. Pour en revenir à l'exemple, la comparaison des 5 mesures s'effectuera donc entre l'attribut de* $\boldsymbol{A}$ *et l'attribut de* $\boldsymbol{B}$ *les plus similaires* selon ces mêmes mesures*, puis sur les seconds plus similaires et ainsi de suite, pour finir par comparer les mesures relevées sur les deux attributs surnuméraires de* $\boldsymbol{A}$ *avec leur valeur sur un hypothétique attribut vide de* $\boldsymbol{B}$*. Cette comparaison par paires permet de s'affranchir de l'ordre de présentation des attributs, qui ne recèle aucune information, se concentrant sur la topologie réelle du jeu de données (Figure 3)).*

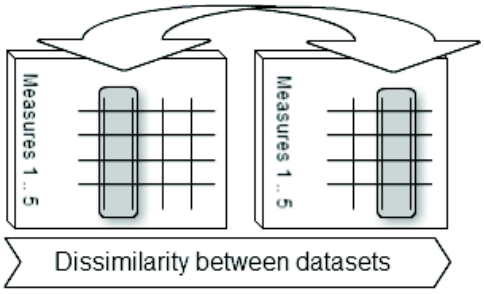

Figure 3: Comparaison directe d'attributs

Cette comparaison peut s'effectuer par le biais d'une dissimilarité prenant en compte les propriétés des attributs individuels des jeux de données.

3. FONCTION DE DISSIMILARITÉ

3.1 Propriétés désirables

Avant de proposer une fonction candidate, il convient d'étudier les propriétés qu'elle devrait présenter. Soit Ω l'ensemble des jeux de données, et $x, x' \in \Omega$ des instances de jeu de données. Traditionnellement, les propriétés usuelles des distances ne sont pas jugées nécessaires [26], ne conservant que la positivité ($d(x,x') \geq 0$). Nous préférerions cependant conserver uniquement des propriétés présentant une interprétation naturelle dans le contexte de la caractérisation de jeu de données.

- $\surd$ Positivité ($d(x,x') \geq 0$) : une dissimilarité doit quantifier la différence absolue entre éléments, donc naturellement positive.
- $\surd$ Indiscernabilité des identiques ($x = x' \rightarrow d(x,x') = 0$) : des jeux de données rigoureusement identiques doivent être considérés aussi similaires que possible.
- $\times$ Identité des indiscernables ($d(x,x') = 0 \rightarrow x = x'$) : des jeux de données physiquement différents doivent pouvoir être considérés parfaitement similaires (considérer par exemple l'ordre de présentation des attributs).
- $\surd$ Symétrie ($d(x,x') = d(x',x)$) : l'ordre de présentation des jeux de données est indifférent, il parait donc naturel de l'ignorer.
- $\times$ Inégalité triangulaire ($d(x,x'') \leq d(x,x') + d(x',x'')$) : perd tout sens en l'absence d'identité des indiscernables. On peut avoir $d(x,x') = d(x',x'') = 0$ et néanmoins $x \neq x''$ et $d(x,x'') \geq 0$...

Définition 1 *Soit A un ensemble et d une fonction de $A^2 \rightarrow \mathbb{R}$. d est une* ***fonction de dissimilarité*** *sur A si et seulement si, $\forall x, x' \in A^2$:*

- $d(x,x') \geq 0$ *(Positivité)*
- $x = x' \rightarrow d(x,x') = 0$ *(Indiscernabilité des identiques)*
- $d(x,x') = d(x',x)$ *(Symétrie)*

Afin de construire une dissimilarité entre jeux de données, il faudra pouvoir comparer les valeurs de différentes propriétés de ces jeux de données. Ces propriétés étant possiblement très différentes les unes des autres de par leur sémantique et représentation, une normalisation sera nécessaire pour garantir qu'aucune composante ne domine les autres ou ne soit ignorée. Ces propriétés sont formalisées dans la définition ci-dessous, où un ensemble de valeurs sera dit *atomique* s'il ne peut être divisé en sous-ensemble dont l'observation indépendante produirait autant d'information que l'observation de l'ensemble complet.

Définition 2 *Soit A un ensemble fini et d une fonction de dissimilarité sur A. d est une* ***fonction de dissimilarité normalisée*** *sur A si et seulement si au moins l'une des propriétés suivantes est vérifiée :*

1. *A est atomique et d est bornée sur A*
2. *Il existe une suite d'ensembles $E_1...E_n$ tels que $A = \prod_{i=1}^{n} E_i$, et une suite de fonctions de dissimilarité $d_1...d_n$ respectivement normalisées sur $E_1...E_n$, telles que :*

$$\forall \delta \in \mathbb{R},\ \exists \Delta \in \mathbb{R} \text{ tel que } \forall i \in [1..n] \text{ et } \forall a,b,c \in A,$$
$$Si\ d_i(a_i,b_i) = d_i(a_i,c_i) + \delta * \max_{(x,y)\in A^2}(d_i(x_i,y_i))$$
$$et\ \forall j \in [1..n],\ j \neq i,\ d_j(a_j,b_j) = d_j(a_j,c_j)$$
$$Alors\ d(a,b) = d(a,c) + \Delta \text{ et } \Delta = 0 \leftrightarrow \delta = 0$$

En d'autres termes, une variation d'amplitude δ relativement à sa borne supérieure, de toute composante d_i entre deux éléments de A induit une même variation Δ de d.

3.2 Fonction candidate

Nous proposerons ici une fonction de dissimilarité particulière présentant les propriétés énoncées précédemment. Pour construire ces fonctions, nous considérerons un ensemble fini de jeux de données ω, et deux ensembles de mesures. Le premier, G, consistera en des propriétés générales de jeu de données, telles que présentées en section 2. Le second, F, consistera en des propriétés capables de caractériser les attributs individuels de jeux de données.

Définition 3 *Soit $E_1...E_n$ une suite d'ensembles finis et A leur produit cartésien $\prod_{i=1}^{n} E_i$. Soit $d_1...d_n$ une suite de fonctions de dissimilarité respectivement sur $E_1...E_n$. On définit la* ***dissimilarité normalisée par la borne supérieure*** *sur A selon $d_1...d_n$, $d_A^{ubr} : A^2 \mapsto \mathbb{R}^+$ telle que*[1] *:*

$$\forall a,b \in A,\ d_A^{ubr}(a,b) = \sum_{i=1}^{n} \frac{d_i(a_i,b_i)}{\max_{(x,y)\in A^2}(d_i(x_i,y_i))} \qquad (1)$$

Proposition 1 *Soit $E_1...E_n$ une suite d'ensembles finis et A leur produit cartésien $\prod_{i=1}^{n} E_i$. Soit $d_1...d_n$ une suite de fonctions de dissimilarité respectivement normalisées sur $E_1...E_n$. Alors, la* ***dissimilarité normalisée par la borne supérieure*** *sur A selon $d_1...d_n$ est une* ***fonction de dissimilarité normalisée*** *sur A.*

Preuve. Soit $\delta \in \mathbb{R}^*$. Les E_i étant des ensembles finis, et les d_i étant des dissimilarités normalisées, $\max_{(x,y)\in A^2} d_i(x_i,y_i)$ existe $\forall i$. Soit alors $(X,Y,Z) \in A^3$ tels que

$$\exists i \in [1..n], d_i(X_i,Y_i) = d_i(X_i,Z_i) + \delta * \max_{(x,y)\in A^2}(d_i(x_i,y_i))$$
$$\forall j \in [1..n],\ j \neq i,\ d_j(X_j,Y_j) = d_j(X_j,Z_j)$$

Ce qui implique $d_A^{ubr}(X,Y) = d_A^{ubr}(X,Z) + \delta$. Ainsi, $\exists \Delta = \delta$, et d_A^{ubr} est bien une **fonction de dissimilarité normalisée** sur A. □

Supposant que l'on puisse construire des fonctions de dissimilarité normalisées sur $G(\omega)$ et $F(\omega)$, on pourrait donc proposer d_ω^{ubr} comme fonction de dissimilarité normalisée sur ω. Afin d'aléger les notations, dans les paragraphes suivants, pour toute fonction H définie sur ω, on notera abusivement $d_H(H(x),H(y)) = d_H(x,y)$.

3.2.1 Méta-attribut des jeux de données

Afin de disposer d'une large sélection de méta-attributs de diverses catégories, nous avons choisi d'utiliser ceux de la base de données d'expériences d'apprentissage OpenML [24]. Les nombreuses expériences présentes sur cette base et dont les données sont caractérisées par cet ensemble de

[1] *ubr* pour *upper bound relative*

méta-attributs constitueront une large base de comparaison. OpenML affiche plus d'une centaine de méta-attributs des jeux de données, provenant de différentes approches statistiques, information-théorétiques, et de "landmarking" (voir `http://www.openml.org/` pour une liste complète). Les valeurs $g(\omega)$ de l'un de ces méta-attributs g sur nos jeux de données constitueront le cas typique d'ensembles *atomiques* à partir desquels lesquels calculer la dissimilarité. On peut donc définir pour chaque méta-attribut g une dissimilarité bornée $d_g : g(\omega)^2 \mapsto \mathbb{R}^+$ (souvent la différence absolue), qui selon la définition 2.1 sera donc normalisée. Ceci permet d'introduire la dissimilarité normalisée par la borne supérieure (cf. Eq. 1) sur $G(\omega)$ selon $\{d_g | g \in G\}$:

$$\forall x, y \in \omega,\ d^{ubr}_{G(\omega)}(x, y) = \sum_{g \in G} \frac{d_g(x, y)}{\max\limits_{(x',y') \in \omega^2} (d_g(x', y'))} \qquad (2)$$

En pratique, cela coïncidera généralement avec une distance de Manhattan normalisée. Cela pose en revanche les fondations nécessaires au prochain type de mesures : les méta-attributs caractérisant les attributs individuels des jeux de données.

3.2.2 Méta-attributs des attributs

Afin de caractériser les attributs individuels des jeux de données, on définit un ensemble de mesures statistiques, information-théorétiques, et de "landmarking", consistant majoritairement en des versions non agrégées des *méta-attributs des jeux de données*. Cet ensemble F de *méta-attributs des attributs* comprend les 72 mesures d'OpenML permettant de caractériser les attributs de jeux de données (voir *Ressources* pour une liste complète). Certaines peuvent caractériser tout type d'attribut (le *nombre de valeurs manquantes*, par exemple), tandis que d'autres sont restreintes à des types particuliers. Dans la définition de ces mesures, nous nous sommes concentrés sur les deux types d'attributs les plus représentés : attributs nominaux et numériques. Les vecteurs de méta-attributs caractérisant les attributs individuels présenteront donc nécessairement des valeurs manquantes (notées $\emptyset$) pour les mesures inadaptées à leur type, ce qui est un obstacle majeur à leur comparaison. En effet, la signification d'une différence de valeur entre deux jeux de données est intrinsèquement dépendante du méta-attribut considéré, et varie grandement d'un méta-attribut à l'autre. Afin de pouvoir comparer la valeur $f(x_i)$ d'un méta-attribut $f \in F$ sur x_i le i^{th} attribut du jeu de données $x \in \omega$, et x'_j le j^{th} attribut du jeu de données $x' \in \omega$, on introduit les fonctions $\delta_f : f(\omega)^2 \mapsto \mathbb{R}^+$ et $\delta^{\emptyset}_f : f(\omega) \mapsto \mathbb{R}^+$ telles que :

1. δ_f est une dissimilarité bornée sur l'ensemble atomique $f(\omega)$ (donc normalisée)
2. $\delta_f(x_i, \emptyset) = \delta_f(\emptyset, x_i) = \delta^{\emptyset}_f(x_i)$
3. $\delta^{\emptyset}_f$ est la *dissimilarité à l'absence de valeur* du méta-attribut f. Elle doit être définie en considérant le *sens* d'une valeur manquante de f, et sera détaillée plus avant par la suite.

On peut alors définir $\delta_F : F(\omega)^2 \mapsto \mathbb{R}^+$:

$$\delta_F(x_i, x'_j) = \sum_{f \in F} \frac{\delta_f(x_i, x'_j)}{\max\limits_{\substack{(y,y') \in \omega^2 \\ (p,q) \in \mathbb{N}^2}} \delta_f(y_p, y'_q)))} \qquad (3)$$

δ_F permet de comparer les attributs de différents jeux de données selon les *Méta-attributs des attributs* de F. Cependant, comme indiqué en section 2 et illustré en Figure 3, l'objectif est de comparer ces attributs par paires les plus similaires. Cela peut se faire de la manière présentée ci-après. Soient n et n' le nombre d'attributs respectifs des jeux de données x et x'. Moyennant une éventuelle substitution, supposons $n \geq n'$. Soit alors S l'ensemble des pseudo-permutations $\sigma : [\![1, n]\!] \to [\![1, n']\!] \cup \{\emptyset\}$ surjectives sur $[\![1, n']\!] \cup \{\emptyset\}$ et injectives sur $[\![1, n']\!]$, ainsi qu'illustré en Figure 4.

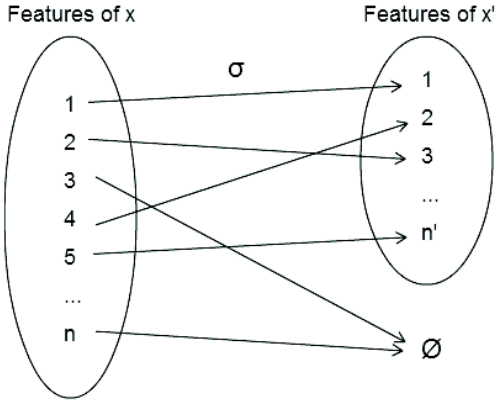

Figure 4: Pseudo-permutation σ

On peut alors définir $d_{F(\omega)} : F(\omega)^2 \mapsto \mathbb{R}^+$ telle que :

$$d_{F(\omega)}(x, x') = \min_{\sigma \in S} \tfrac{1}{n} \sum_{i=1}^{n} \delta_F(x_i, x'_{\sigma(i)})$$

$$d_{F(\omega)}(x, x') = \min_{\sigma \in S} \frac{1}{n} \sum_{i=1}^{n} \sum_{f \in F} \frac{\delta_f(x_i, x'_{\sigma(i)})}{\max\limits_{\substack{(y,y') \in \omega^2 \\ (p,q) \in \mathbb{N}^2}} \delta_f(y_p, y'_q)))} \qquad (4)$$

Proposition 2 *$d_{F(\omega)}$ est une fonction de dissimilarité normalisée sur $F(\omega)$.*

Preuve. On démontre successivement quatre propriétés : Positivité, Indiscernabilité des identiques, Symétrie et Normalisation (au sens de la Définition 2).

1. Soient $x, x' \in \omega$ possédant respectivement n et n' attributs. Montrons que $d_{F(\omega)}(x, x') \geq 0$.
   $\forall f \in F$, δ_f est une fonction de dissimilarité, donc $\forall i, j \in [1, n] * [1, n']$, $\delta_f(x_i, x'_j) \geq 0$, et par somme, $d_{F(\omega)}(x, x') \geq 0$. ■

2. Soit $x \in \omega$ ayant n attributs. Montrons $d_{F(\omega)}(x, x) = 0$.

   $$d_{F(\omega)}(x, x) = \min_{\sigma \in S} \tfrac{1}{n} \sum_{i=1}^{n} \sum_{f \in F} \frac{\delta_f(x_i, x_{\sigma(i)})}{\max\limits_{\substack{(y,y') \in \omega^2 \\ (p,q) \in \mathbb{N}^2}} \delta_f(y_p, y'_q)))}$$

   $\forall f \in F$, δ_f est une fonction de dissimilarité, donc $\forall i, j \in [1, n]$, $\delta_f(x_i, x_j) \geq 0$, et par somme,

   $$\tfrac{1}{n} \sum_{i=1}^{n} \sum_{f \in F} \frac{\delta_f(x_i, x_{\sigma(i)})}{\max\limits_{\substack{(y,y') \in \omega^2 \\ (p,q) \in \mathbb{N}^2}} \delta_f(y_p, y'_q)))} \geq 0$$

De plus, comme δ_f est une fonction de dissimilarité, $\forall i \in [1,n]$, $\delta_f(x_i,x_i) = 0$. Donc, pour $\sigma = Id$ la fonction Identité, on a :

$$\frac{1}{n}\sum_{i=1}^{n}\sum_{f\in F}\frac{\delta_f(x_i,x_{\sigma(i)})}{\max\limits_{\substack{(y,y')\in\omega^2\\(p,q)\in\mathbb{N}^2}}\delta_f(y_p,y'_q)))} = 0$$

$d_{F(\omega)}(x,x)$ étant un minimum selon σ, on a donc bien $d_{F(\omega)}(x,x) = 0$. ■

3. Soient $x, x' \in \omega$ possédant respectivement n et n' attributs. Montrons que $d_{F(\omega)}(x,x') = d_{F(\omega)}(x',x)$.
Soit $m = max(n,n')$. $\forall f \in F$, δ_f est une fonction de dissimilarité, donc $\forall i,j \in [1,n] * [1,n']$,

$$\delta_f(x_i,x'_j) = \delta_f(x'_j,x_i)$$

$$\frac{\delta_f(x_i,x'_j)}{\max\limits_{\substack{(y,y')\in\omega^2\\(p,q)\in\mathbb{N}^2}}\delta_f(y_p,y'_q)))} = \frac{\delta_f(x'_j,x_i)}{\max\limits_{\substack{(y,y')\in\omega^2\\(p,q)\in\mathbb{N}^2}}\delta_f(y_p,y'_q)))}$$

$$\min_{\sigma\in S}\frac{1}{m}\sum_{i=1}^{m}\sum_{f\in F}\frac{\delta_f(x_i,x'_j)}{\max\limits_{\substack{(y,y')\in\omega^2\\(p,q)\in\mathbb{N}^2}}\delta_f(y_p,y'_q)))} =$$

$$\min_{\sigma\in S}\frac{1}{m}\sum_{i=1}^{m}\sum_{f\in F}\frac{\delta_f(x'_j,x_i)}{\max\limits_{\substack{(y,y')\in\omega^2\\(p,q)\in\mathbb{N}^2}}\delta_f(y_p,y'_q)))}$$

$$d_{F(\omega)}(x,x') = d_{F(\omega)}(x',x) \quad ■$$

4. $F(\omega)$ ne peut être partitionné sans perte d'information (typiquement, le nombre d'attributs du jeu de données), et sera donc dit *atomique*. $F(\omega)$ est un ensemble fini, donc $d_{F(\omega)}$ est bornée sur $F(\omega)$. Selon la Définition 2.1, $d_{F(\omega)}$ est donc normalisée sur $F(\omega)$. ■

$d_{F(\omega)}$ est donc bien une fonction de dissimilarité normalisée sur $F(\omega)$. □

3.2.3 *Composition.*

Par le biais des équations 2 et 4, on peut donc proposer $\forall x,y \in \omega$ la **dissimilarité normalisée par la borne supérieure** sur ω selon $d^{ubr}_{G(\omega)}$ et $d_{F(\omega)}$:

$$d^{ubr}_{\omega}(x,y) = \frac{d^{ubr}_{G(\omega)}(x,y)}{\max\limits_{(x',y')\in\omega^2}(d^{ubr}_{G(\omega)}(x',y'))} + \frac{d_{F(\omega)}(x,y)}{\max\limits_{(x',y')\in\omega^2}(d_{F(\omega)}(x',y'))} \quad (5)$$

D'après la Proposition 1, d^{ubr}_{ω} est bien une fonction de dissimilarité normalisée sur ω.

4. VALIDATION

Afin d'évaluer l'utilité de la dissimilarité proposée, cette section présentera une procédure de validation en deux étapes. Tout d'abord, nous étudierons le potentiel de la dissimilarité proposée pour l'apprentissage, en utilisant les méthodes d'estimation présentées dans [26]. Dans un second temps, nous décrirons une importante expérience de méta-apprentissage évaluant la dissimilarité proposée dans un large panel de scénarios.

Ces deux approches requerront la définition d'un problème précis de méta-apprentissage comme cas d'étude, et la collecte de données caractérisant ce problème. Nous avons concentré nos efforts sur les problèmes de classification. En effet, s'agissant du cas le plus typique de l'apprentissage, il est notablement plus facile de collecter des expériences de classification. La description de ces expériences proviendra également de la base d'OpenML.

4.1 Validation Théorique

Dans [26], Wang & al. proposent une définition intuitive de la qualité d'une fonction de dissimilarité dans le contexte de l'apprentissage. Selon leur définition, une fonction de dissimilarité est *strongly* (ϵ,γ)*-good* pour un problème donné de classification binaire, si une proportion au moins $1-\epsilon$ des exemples $z = (x,y)$ satisfait :

$$P(d(x,x') < d(x,x'') \mid y' = y,\ y'' = -y) \geq \frac{1}{2} + \frac{\gamma}{2}$$

En d'autres termes, plus la probabilité que la dissimilarité juge deux exemples aléatoires de même classe plus proches que deux exemples aléatoires de classes différentes est grande, mieux elle permettra de séparer les classes. Cette interprétation nous amène à définir un problème de classification binaire entrant dans les attributions de la dissimilarité proposée.

Considérons un ensemble X de jeux de données de classification, et un ensemble A de classifieurs. On exécute chaque classifieur de A sur chaque jeu de données de X et mesure un critère de performance c du modèle résultant. Ensuite, pour chaque jeu de données $x \in X$, on définit l'ensemble A_x des algorithmes *appropriés* sur ce jeu de données selon le critère de performance c, comme ceux étant au plus un écart type en dessous du meilleur :

$$A_x = \{a \in A \text{ tels que } |\max_{a'\in A}(c(a',x)) - c(a,x)| \leq \sigma_x\}$$

On peut donc considérer, pour tout algorithme $a \in A$, le problème de classification binaire où les instances sont les jeux de données $x \in X$, et dont la classe spécifie si a est approprié sur x. Ces problèmes caractérisent donc l'adéquation entre les différents classifieurs et jeux de données, ce qui est un objectif intuitif de la dissimilarité proposée.

Pour évaluer la dissimilarité, on peut alors calculer pour chaque classifieur $a \in A$ et chaque jeu de données $x \in X$, la probabilité que la dissimilarité juge deux exemples aléatoires de même classe plus proches que deux exemples aléatoires de classes différentes, ce qui nous donnera la (ϵ,γ)*-goodness* de d^{ubr}_{ω} :

$$P(d^{ubr}_{\omega}(x,x') < d^{ubr}_{\omega}(x,x'') \mid a \in A_x,\ a \in A_{x'},\ a \notin A_{x''})$$

Le résultat suivant de [26], stipule que si d est une fonction de dissimilarité *strongly* (ϵ,γ)*-good*, alors il existe un classifieur simple construit sur d qui, sur le choix de $n = \frac{4}{\gamma^2} ln \frac{1}{\delta}$ paires d'exemples aléatoires de classes différentes, aura, avec une probabilité au moins $1-\delta$, un taux d'erreur d'au plus $\epsilon + \delta$. Ce résultat permet d'estimer de manière naturelle l'adéquation de la dissimilarité au problème étudié.

Ces mesures ont été réalisées sur des ensembles de classifieurs et de jeux de données provenant de la base d'OpenML (voir *Ressources* pour les tables complètes), en utilisant successivement la dissimilarité proposée, une distance euclidienne, et une distance de Manhattan sur les méta-attributs des jeux de données. Le paramètre γ a été amené aussi haut que possible en conservant $\epsilon \leq 0.05$. Les résultats sont présentés en Table 1, moyennés selon les classifieurs et jeux de données. La Table 2 présente le risque δ et la borne de taux d'erreur obtenus pour différents nombres d'exemples.

	200 exemples		1000 exemples		5000 exemples	
	δ	erreur max	δ	erreur max	δ	erreur max
d_ω^{ubr}	0,973	1,023	0,871	0,921	0,501	0,551
Distance Euclidienne	0,989	1,039	0,945	0,995	0,755	0,805
Distance de Manhattan	0,990	1,040	0,952	1,002	0,783	0,833

	10000 exemples		20000 exemples		50000 exemples	
	δ	erreur max	δ	erreur max	δ	erreur max
d_ω^{ubr}	0,251	0,301	0,063	0,113	0,001	0,051
Distance Euclidienne	0,570	0,620	0,325	0,375	0,060	0,110
Distance de Manhattan	0,613	0,663	0,375	0,425	0,086	0,136

Table 2: Borne du taux d'erreur obtenu avec une probabilité $1 - \delta$ pour différents nombres d'exemples et par des classifieurs construits sur différentes dissimilarités.

Dissimilarité	ϵ	γ
d_ω^{ubr}	0,05	0,024
Distance Euclidienne	0,05	0,015
Distance de Manhattan	0,05	0,014

Table 1: (ϵ, γ)-*goodness* moyenne avec différentes dissimilarités

Comme on peut le voir, la dissimilarité proposée offre des bornes intéressantes au taux d'erreur avec sensiblement moins d'exemples (environ 10% pour 20k exemples, ce qui n'est atteint par la distance euclidienne qu'autour de 50k...). Elle semble donc être plus adaptée à la caractérisation d'adéquation entre classifieur et jeu de données que les autres distances étudiées. Ce résultat est en revanche nécessairement dépendant du choix des algorithmes et jeux de données sur lesquels sont construits ces problèmes d'adéquation, et on ne peut donc pas le supposer généralisable. Ce qui est montré ici, est que pour *certains* algorithmes, l'utilisation de la dissimilarité proposée permet de caractériser leur adéquation aux jeux de données plus efficacement que les distances traditionnelles. Parmi les algorithmes où la dissimilarité présente d'excellentes performances, on peut noter une majorité de classifieurs de type arbre de décision. On pourrait donc postuler que la dissimilarité caractérise bien l'adéquation des approches par arbres de décision aux jeux de données, et donc que cette adéquation dépend largement des méta-attributs d'attributs individuels utilisés par d_ω^{ubr}.

4.2 Validation Expérimentale

Afin d'évaluer l'intérêt de la dissimilarité proposée dans le cadre de la sélection d'algorithmes et du méta-apprentissage, nous avons mis en place une série d'expériences comparatives autour du problème de méta-classification. Nous illustrerons tout d'abord ce problème par un exemple d'exécution, puis détaillerons les facteurs qui varieront au fil des exécutions.

4.2.1 Exemple d'exécution

Pour un jeu de données de classification particulier, l'objectif d'une expérience de sélection d'algorithmes est de choisir un classifieur maximisant un critère donné. Dans cet exemple, on utilisera le critère traditionnel de précision (bien que biaisé [12], il est l'un des plus intuitifs). Pour construire le jeu de données de méta-classification, on extrait de la base d'OpenML deux ensembles de données. Le premier décrit la précision de différents algorithmes de classification sur un ensemble de jeux de données (Table 5), tandis que le second caractérise ces jeux de données selon un ensemble de méta-attributs (Table 3).

	$classifier_1$	$classifier_2$	...	$classifier_{93}$
$dataset_1$	0.8	0.9	...	...
$dataset_2$	0.9	0.7	...	...
...	...	...	...	...
$dataset_{434}$	...	...	...	...

Table 5: Précision de différent algorithmes de classification sur un ensemble de jeux de données

	$NumberOfInstances$	$NumberOfFeatures$	...	$MetaAttribute_{105}$
$dataset_1$	100	62	...	...
$dataset_2$	5000	13	...	...
...	...	...	...	...
$dataset_{434}$	360	20	...	...

Table 3: Caractérisation des jeux de données selon un ensemble de méta-attributs

	$NumberOfInstances$	...	$MetaAttribute_{105}$	$Class$
$dataset_1$	100	...	4	$classifier_{18}$
$dataset_2$	5000	...	92	$classifier_7$
...	...	...	...	...
$dataset_{434}$	360	...	13	$classifier_{63}$

Table 4: Jeu de données de méta-classification

On peut alors construire le jeu de données de méta-classification représenté en Table 4. La classe d'une instance de ce jeu de données de méta-classification identifie quel algorithme présente la meilleure performance (*i.e.* la meilleure précision) sur le jeu de données qu'elle décrit (selon les performances recensées en Table 5).

Il faut ensuite résoudre ce problème de méta-classification. Dans cet exemple ceci s'effectue par une forme de *"leave one out"* selon le pseudocode 1.

foreach *Instance de jeu de données* $dataset_i$ **do**
- Exclure $dataset_i$ du jeu de données de méta-classification
- Appliquer l'algorithme de sélection d'attributs ReflieF au jeu de données de méta-classification
- Aprendre un arbre de décision en appliquant l'algorithme C4.5 au jeu de données de méta-classification réduit
- Utiliser cet arbre de classification pour prédire la classe de l'instance $dataset_i$

Pseudocode 1: Exemple d'exécution

Pour chaque jeu de données, on dispose alors d'un label de classe prédit, identifiant quel classifieur devrait y obtenir les meilleures performances, selon l'arbre de décision construit sur les autres instances. L'objectif est ensuite de caractériser la performance de cette expérience à partir de cet ensemble de prédictions. Pour ce faire, il convient d'utiliser un critère *aussi indépendant que possible* de tous les choix de construction fait dans l'expérience. Le critère suivant permet de s'en affranchir dans une certaine mesure :

Définition 4 *Soit* $\boldsymbol{p}$ *la performance du classifieur* $\boldsymbol{classifier}_j$ *sur le jeu de données* $\boldsymbol{dataset}_i$ *selon le critère choisi (ici, la précison). Soient alors* $\boldsymbol{best}$ *la performance du meilleur classifieur de* $\boldsymbol{classifier}_{1..m}$ *sur le jeu de données* $\boldsymbol{dataset}_i$*, et* $\boldsymbol{def}$ *la performance du classifieur par défaut (prédisant la classe majoritaire) sur ce même jeu de données. On définit la performance* $\boldsymbol{perf}$ *de notre expérience sur le jeu de données* $\boldsymbol{dataset}_i$ *:*

$$perf = 1 - \frac{|best - p|}{|best - def|}$$

Ce critère de performance atteint son maximum de 1 quand le classifieur prédit présente une valeur de précison maximale, et atteint 0 quand le classifieur prédit présente la même précision que le classifieur par défaut. Le cas négatif traduit une performance inférieure à celle du classifieur par défaut. Bien que simple, ce critère permet de comparer la performance d'expériences construites sur différents méta-problèmes, mais il reste nécessaire de lui fournir la valeur neutre du critère d'évaluation considéré.

4.2.2 Cadre d'expérimentation

Comme stipulé précédemment, le jeu de données de méta-classification est construit en utilisant des données d'OpenML, qui répertorie plus de 2500 jeux de données et 2000 algorithmes. La construction du jeu de données de méta-classification requérant de nombreux algorithmes évalués sur un ensemble de jeux de données, nous avons employé une technique de recherche de bi-clique maximale [23] pour trouver les plus grands ensembles de jeux de données et de classifieurs tels que chaque élément des deux ensembles a été évalué en conjonction avec tous les éléments de l'autre ensemble. Ceci nous a permis de nous restreindre à 93 algorithmes de classification et 434 jeux de données de la base d'OpenML. Nous avons ensuite extrait les valeurs d'évaluation de 11 critères de performance sur chacune des 40k exécutions que cela représente. Enfin nous avons extrait les valeurs de 105 méta-attributs pour chaque jeu de données (voir *Ressources* pour les noms et descriptions de tous ces éléments). Ces méta-attributs constituent également l'ensemble décrit en Section 3.2 et utilisé par la dissimilarité.

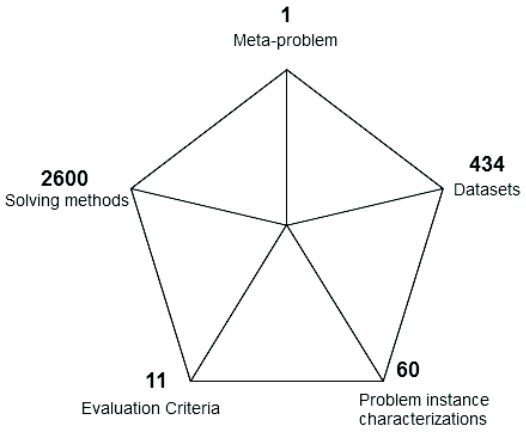

Figure 5: Dimensions des exécutions

Pour itérer l'expérience présentée dans l'exemple précédent, il est ensuite nécessaire de définir des ensembles d'algorithmes de classification et de sélection d'attributs pour le méta-niveau. Nous avons choisi d'utiliser les algorithmes de l'API *Weka* [6], qui de par son statut de référence, bénéficie d'implémentations de nombreux algorithmes de l'état de l'art. Nous évaluons ainsi plus de 2600 classifieurs et 60 méthodes de sélection d'attributs de l'API Weka (voir *Ressources* pour les listes complètes), comprenant en particulier des méthodes d'augmentation

telles le "boosting" ou "bagging". À cela, nous ajoutons un classifieur kNN (IBk dans Weka) employant la dissimilarité proposée comme distance entre instances, et permettons à ce classifieur les mêmes déclinaisons d'hyperparamètres et méthodes d'augmentation que les autres classifieurs.

Les exécutions individuelles sont instanciées automatiquement, et déléguées à un répartiteur de tâches SLURM [30] gérant les 640 nœuds du cluster *OSIRIM* (voir `osirim.irit.fr`). Les 200k exécutions résultantes totalisent plus de 700 millions de cycles apprentissage-prédiction-évaluations, ce qui, même avec des ressources importantes, reste très couteux (plus d'un mois de temps réel en exécution parallèle, pour des dizaines d'années de temps machine).

4.2.3 Résultats

Ce procédé fournit donc une valeur de performance pour chacune des 200k exécutions. Notre objectif étant de discriminer entre classifieurs au méta-niveau, et les valeurs de performance étant commensurables, il est possible d'en faire la moyenne selon les autres dimensions. On obtient donc, pour chaque classifieur de méta-niveau, sa performance moyenne pour différentes approches de sélection d'attributs, critères d'évaluation à maximiser, et ensembles d'apprentissage. On minimise ainsi la variance introduite par ces différents facteurs de bruit. Notre jeu de données de méta-classification étant néanmoins de taille relativement petite (434 instances), on ne peut complètement négliger la variance de ces résultats, et il conviendra donc de les étudier dans leur globalité, car plus on "zoome" sur les résultats, plus on a de chances d'observer un résultat contingent, fonction uniquement du contexte de l'expérience. Nous avons donc concentré nos observations sur la répartition des classifieurs de méta-niveau utilisant la dissimilarité proposée dans la population. Dans la Figure 6, on compare, pour différents seuils de performance, le nombre de classifieurs de méta-niveau utilisant ou non la dissimilarité proposée. On peut constater que parmi les approches conseillant en moyenne des classifieurs jusqu'à 5% moins performant que le meilleur, 20% sont basées sur la dissimilarité proposée, alors que cette proportion est de moins de 10% dans la population complète.

La Figure 7 présente la proportion de classifieurs de méta-niveau utilisant la dissimilarité proposée dans chaque décile de performance. Là encore, on peut noter que parmi les 10% de meilleurs classifieurs de méta-niveau, plus de 17% utilisent la dissimilarité proposée. De plus, malgré des divergences aux deuxième, quatrième et sixième déciles, on peut observer que cette proportion semble décroître presque linéairement avec la baisse de performance.

Ces résultats ne permettent pas d'établir la supériorité générale des approches employant la dissimilarité proposée, mais montrent bien que certaines approches en bénéficient grandement. Ce constat n'est pas particulièrement surprenant dans la mesure où le méta-apprentissage est toujours concerné par les limitations des *"no free lunch theorems"* [28, 29]. En effet, toute nouvelle méthode peut, au mieux, faire montre de meilleures performances sur une partie spécifique du problème. Parmi les méthodes bénéficiant le plus de l'utilisation de la dissimilarité, on peut relever une majorité d'approches divisant le problème en plusieurs sous-problèmes de classification binaire, telles *"ensemblesOfNestedDichotomies"* [2] ou *"ClassificationViaRegression"* [3]. Ceci pourrait suggérer que la dissimilarité proposée est particulièrement efficace sur de tels sous-problèmes, qui reviennent à caractériser les domaines de bonne performance d'algorithmes particuliers. Il s'agit là de l'une des problématiques récurrentes de l'apprentissage, et donc d'un développement potentiel intéressant pour la dissimilarité proposée.

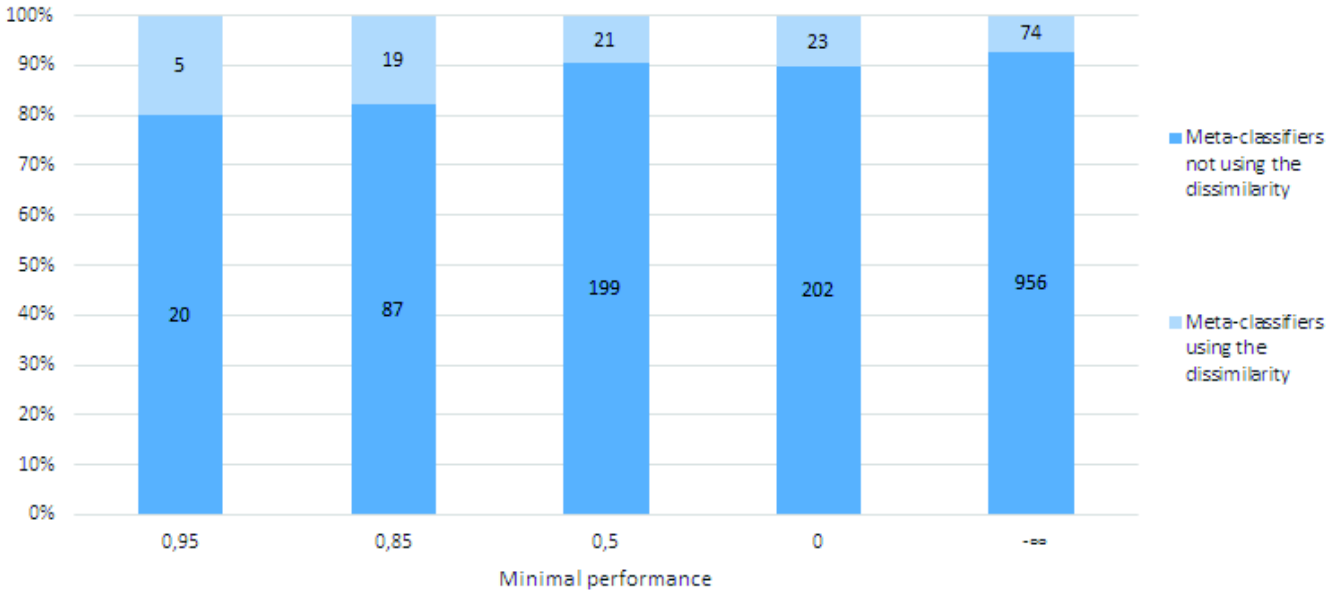

Figure 6: Populations de méta-classifieurs atteignant divers seuils de performance

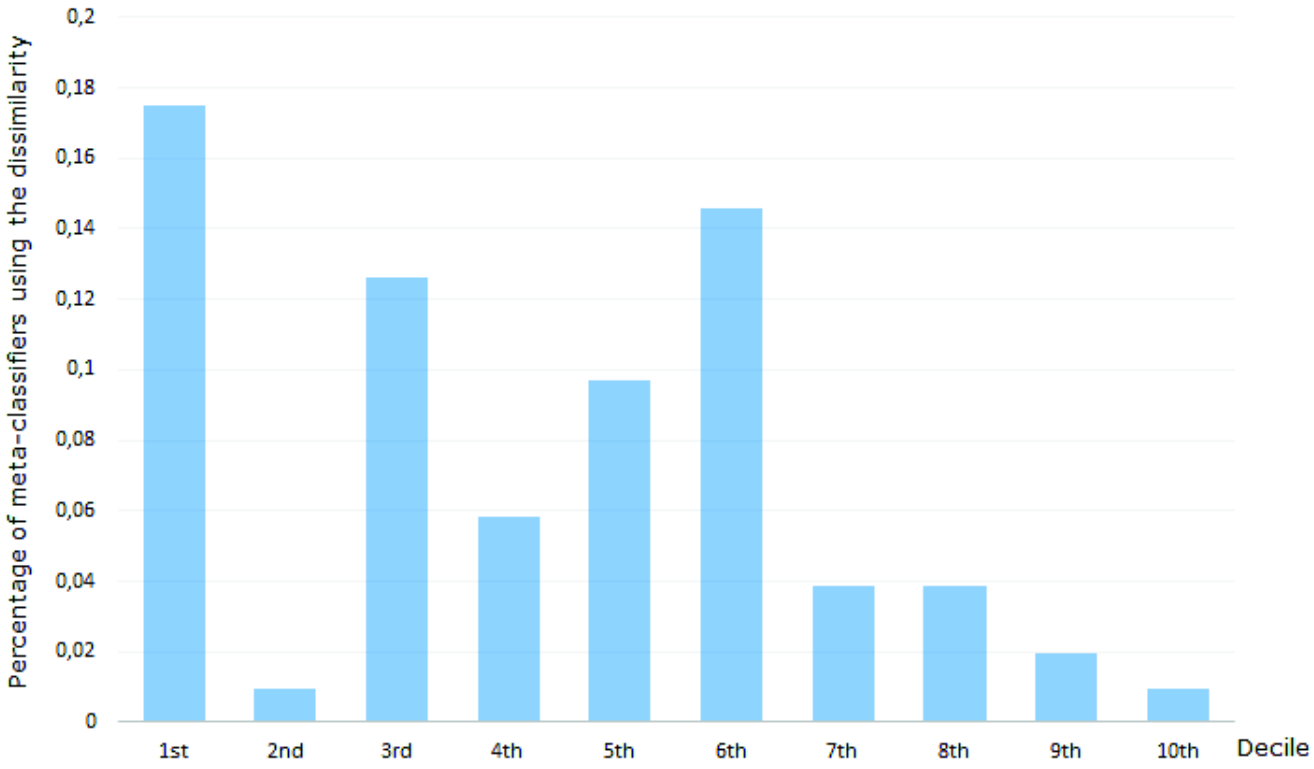

Figure 7: Proportion dans chaque décile de performance de méta-classifieurs utilisant la dissimilarité proposée

5. CONCLUSION

Nous avons proposé une fonction de dissimilarité entre jeux de données présentant un ensemble de propriétés désirables, et capable d'employer des méta-attributs caractérisant des attributs particuliers de ces jeux de données. Nous avons montré qu'elle permet de caractériser l'adéquation d'algorithmes de classification avec des jeux de données plus efficacement que des distances traditionnelles, et qu'elle peut être employée avec de bonnes performances dans le contexte de classification au méta-niveau.

De nombreuses pistes d'amélioration restent cependant à explorer. Tout d'abord, notre dissimilarité permet d'utiliser des méta-attributs caractérisant des attributs particuliers des jeux de données, mais diverses expériences [17, 1] ont montré que les propriétés d'un attribut *dans le contexte des autres attributs* sont au moins aussi importantes. Il serait alors intéressant de permettre l'utilisation de tels méta-attributs relationnels, comme la covariance, ou l'information mutuelle, par la dissimilarité. D'autre part, bien que divers et provenant d'approches très différentes, les méta-attributs employés dans nos expériences ne couvrent pas complètement l'état de l'art en la matière. Ces dernières années ont été riches en contributions introduisant de nouveaux méta-attributs [18, 7, 16, 21], dont l'utilisation pourrait révéler l'intérêt d'une approche par dissimilarité dans de nouveaux contextes. Enfin, comme l'efficacité de l'approche par dissimilarité apparaît très dépendante du contexte (comme c'est souvent le cas en apprentissage et méta-apprentissage), il pourrait être intéressant de concevoir une méthode d'évaluation de méta-attributs considérant leurs diverses natures (globaux, liés à un attribut, relationnels...). Il serait alors possible de caractériser l'utilité des divers méta-attributs dans une variété de situations et donc d'approfondir notre connaissance du problème de méta-apprentissage.

Dans le cadre de l'assistance intelligente à l'analyse de données, un atout particulier de notre approche est qu'elle permettrait une caractérisation unifiée des expériences d'analyse de données. En effet, disposant d'une quelconque représentation du processus d'analyse de données et de ses résultats, il serait possible de l'intégrer dans la dissimilarité, permettant la comparaison directe d'expériences complètes. Il s'agit là d'un premier pas vers de nouvelle approches d'assistance intelligente à l'analyse de données, permettant notamment l'utilisation directe d'heuristiques pour la découverte et recommandation de processus d'analyse adaptés.

Ressources. Tous les éléments référencés sont disponibles au téléchargement à l'adresse suivante :
`https://github.com/WilliamR03/Dissimilarity-dataset-characterization.git`

6. REFERENCES

[1] Brown, G., Pocock, A., Zhao, M.J., Luján, M.: Conditional likelihood maximisation: a unifying framework for information theoretic feature selection. The Journal of Machine Learning Research 13(1), 27–66 (2012)

[2] Dong, L., Frank, E., Kramer, S.: Ensembles of balanced nested dichotomies for multi-class problems. In: Knowledge Discovery in Databases: PKDD 2005, pp. 84–95. Springer (2005)

[3] Frank, E., Wang, Y., Inglis, S., Holmes, G., Witten, I.H.: Using model trees for classification. Machine Learning 32(1), 63–76 (1998)

[4] Fürnkranz, J., Petrak, J.: Extended data characteristics. Tech. rep., METAL consortium (2002), accessed 12/11/15 at citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.97.302

[5] Giraud-Carrier, C., Vilalta, R., Brazdil, P.: Introduction to the special issue on meta-learning. Machine learning 54(3), 187–193 (2004)

[6] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. ACM SIGKDD explorations newsletter 11(1), 10–18 (2009)

[7] Ho, T.K., Basu, M.: Complexity measures of supervised classification problems. Pattern Analysis and Machine Intelligence, IEEE Transactions on 24(3), 289–300 (2002)

[8] Kalousis, A.: Algorithm selection via meta-learning. Ph.D. thesis, Universite de Geneve (2002)

[9] Kalousis, A., Gama, J., Hilario, M.: On data and algorithms: Understanding inductive performance. Machine Learning 54(3), 275–312 (2004)

[10] Kalousis, A., Hilario, M.: Model selection via meta-learning: a comparative study. International Journal on Artificial Intelligence Tools 10(04), 525–554 (2001)

[11] Kalousis, A., Hilario, M.: Feature selection for meta-learning. In: Proceedings of the 5th Pacific-Asia Conference on Knowledge Discovery and Data Mining. pp. 222–233. PAKDD '01, Springer-Verlag, London, UK, UK (2001), http://dl.acm.org/citation.cfm?id=646419.693650

[12] Kononenko, I., Bratko, I.: Information-based evaluation criterion for classifier's performance. Mach. Learn. 6(1), 67–80 (Jan 1991)

[13] Leite, R., Brazdil, P., Vanschoren, J.: Selecting classification algorithms with active testing. In: Machine Learning and Data Mining in Pattern Recognition, pp. 117–131. Springer (2012)

[14] Leyva, E., Gonzalez, A., Perez, R.: A set of complexity measures designed for applying meta-learning to instance selection. Knowledge and Data Engineering, IEEE Transactions on 27(2), 354–367 (2015)

[15] Michie, D., Spiegelhalter, D.J., Taylor, C.C.: Machine Learning, Neural and Statistical Classification. Ellis Horwood, Upper Saddle River, NJ, USA (1994)

[16] Ntoutsi, I., Kalousis, A., Theodoridis, Y.: A general framework for estimating similarity of datasets and decision trees: exploring semantic similarity of decision trees. In: SDM. pp. 810–821. SIAM (2008)

[17] Peng, H., Long, F., Ding, C.: Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. Pattern Analysis and Machine Intelligence, IEEE Transactions on 27(8), 1226–1238 (2005)

[18] Peng, Y., Flach, P.A., Brazdil, P., Soares, C.: Decision tree-based data characterization for meta-learning. IDDM-2002 p. 111 (2002)

[19] Pfahringer, B., Bensusan, H., Giraud-Carrier, C.: Tell me who can learn you and i can tell you who you are: Landmarking various learning algorithms. In: Proceedings of the 17th international conference on machine learning. pp. 743–750 (2000)

[20] Serban, F.: Toward effective support for data mining using intelligent discovery assistance. Ph.D. thesis (2013)

[21] Sun, Q., Pfahringer, B.: Pairwise meta-rules for better meta-learning-based algorithm ranking. Machine learning 93(1), 141–161 (2013)

[22] Todorovski, L., Brazdil, P., Soares, C.: Report on the experiments with feature selection in meta-level learning. In: Proceedings of the PKDD-00 workshop on data mining, decision support, meta-learning and ILP: forum for practical problem presentation and prospective solutions. pp. 27–39. Citeseer (2000)

[23] Uno, T., Asai, T., Uchida, Y., Arimura, H.: An efficient algorithm for enumerating closed patterns in transaction databases. In: Discovery science. pp. 16–31 (2004)

[24] Vanschoren, J., van Rijn, J.N., Bischl, B., Torgo, L.: Openml: Networked science in machine learning. SIGKDD Explorations 15(2), 49–60 (2013), http://doi.acm.org/10.1145/2641190.2641198

[25] Vilalta, R., Drissi, Y.: A perspective view and survey of meta-learning. Artif. Intell. Rev. 18(2), 77–95 (Oct 2002), http://dx.doi.org/10.1023/A:1019956318069

[26] Wang, L., Sugiyama, M., Yang, C., Hatano, K., Feng, J.: Theory and algorithm for learning with dissimilarity functions. Neural computation 21(5), 1459–1484 (2009)

[27] Wistuba, M., Schilling, N., Schmidt-Thieme, L.: Learning data set similarities for hyperparameter optimization initializations. pp. 15–26. http://ceur-ws.org/Vol-1455/#paper-04

[28] Wolpert, D.H.: The lack of a priori distinctions between learning algorithms. Neural computation 8(7), 1341–1390 (1996)

[29] Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. Evolutionary Computation, IEEE Transactions on 1(1), 67–82 (1997)

[30] Yoo, A.B., Jette, M.A., Grondona, M.: Slurm: Simple linux utility for resource management. In: Job Scheduling Strategies for Parallel Processing. pp. 44–60. Springer (2003)

[31] Zakova, M., Kremen, P., Zelezny, F., Lavrac, N.: Automating knowledge discovery workflow composition through ontology-based planning. Automation Science and Engineering, IEEE Transactions on 8(2), 253–264 (2011)